LA-UR-02-6148

Approved for public release;
distribution is unlimited.

Title:

A Direct Eulerian Method for Three-Dimensional AMR Radiation-Hydrodynamic Codes (U)

Author(s):

Michael R. Clover
Michael L. Gittings

Submitted to:

Nuclear Explosives Code Developers' Conference
NECDC 2002
October 21–24
Monterey, California

A Direct Eulerian Method for Three-Dimensional AMR Radiation-Hydrodynamic Codes (LA-UR-02-6148)

Michael Clover (SAIC, X-2)
Michael Gittings (SAIC, X-2)

23 October 2002

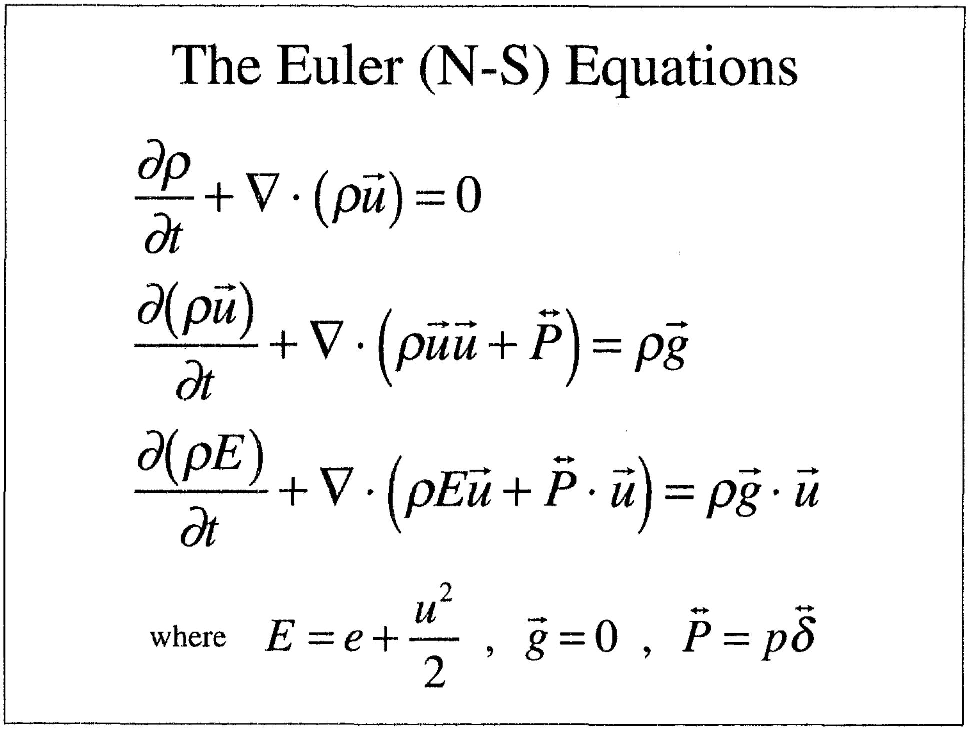
Abstract:

We present a new Direct Eulerian method for hydrodynamics that uses as much "Lagrangian" philosophy as possible -- on the principle that we want a hydrodynamics method to be "as Lagrangian as possible, as long as possible" (e.g. in order to do adiabatic compression exactly).

We perform Strang-splitting in 2-d and 3-d, which is known to be 2nd order accurate in smooth flows. For the 1-d sweeps, we solve the Riemann problem in the fluid frame at each face of our Eulerian mesh. This generates wave characteristics, which are used to infer at what point a Lagrangian mass-point would start - in the donor zone - in order to just reach the face at the end of a timestep. Given linear (re)constructions of all fluid quantities, we advect exactly with a 3-point Gaussian quadrature.

For multi-fluid problems, we convolve the fractional density (frac_mass/frac_vol) with the fractional volume to determine advection of each component. This allows an interface reconstructor to be invoked to return "tweaked" volume fractions for advection.

We present numerous examples to show the order of accuracy of the method.



The Euler (N-S) Equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$$

$$\frac{\partial (\rho \vec{u})}{\partial t} + \nabla \cdot \left(\rho \vec{u}\vec{u} + \overleftrightarrow{P} \right) = \rho \vec{g}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot \left(\rho E \vec{u} + \overleftrightarrow{P} \cdot \vec{u} \right) = \rho \vec{g} \cdot \vec{u}$$

where $E = e + \dfrac{u^2}{2}$, $\vec{g} = 0$, $\overleftrightarrow{P} = p \overleftrightarrow{\delta}$

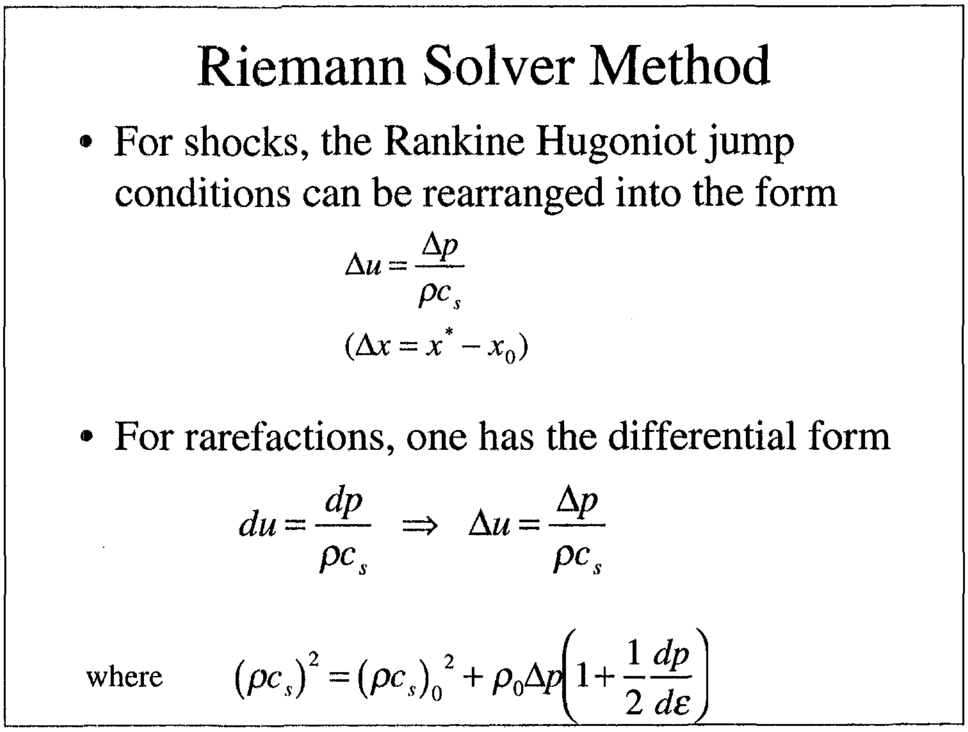
Constant gravity exists in the code at this writing, but has not been significantly tested to date.

Development is ongoing to develop self-gravity. The Poisson equation solver has been developed, but the exact manner of differencing in such a case remains to be determined.

We are also beginning to implement the handling of deviatoric stresses, but this has only been tested in 1-d to date. This will ultimately allow us to calculate with strength of materials as well as real (and turbulent) viscosities.

Thermal conduction is done via operator-splitting, later in the code.

We make no effort to go beyond Strang-splitting to solve these equations because every other physics package is also operator-split, and therefore nothing will be second-order accurate in time except for analytic, pure hydro, test problems.



Riemann Solver Method

- For shocks, the Rankine Hugoniot jump conditions can be rearranged into the form

$$\Delta u = \frac{\Delta p}{\rho c_s}$$

$$(\Delta x = x^* - x_0)$$

- For rarefactions, one has the differential form

$$du = \frac{dp}{\rho c_s} \quad \Rightarrow \quad \Delta u = \frac{\Delta p}{\rho c_s}$$

where $$(\rho c_s)^2 = (\rho c_s)_0^{\,2} + \rho_0 \Delta p\left(1 + \frac{1}{2}\frac{dp}{d\varepsilon}\right)$$

In order to make the method as second-order as possible, we calculate the jump across a face by extrapolating cell-centered values to the faces using limited slopes.

We use the Lagrangian form of the equations of motion to take those face-centered, beginning-of-timestep values to the half timestep.

The solution of the pair of jump conditions across a face can be done more-or-less accurately, depending on the type of condition. The single intermediate state approximate Riemann solver (similar to the artificial viscosity method of Lagrangian codes) predicts an interface pressure. If this pressure is intermediate between the two initial pressures, we calculate the Riemann velocity with the same approximation. Otherwise, we use a Newton iteration method to solve self-consistently for the velocity and sound-speeds as shown above (usually only necessary if two fluids smash into each other, or if they are jerked apart).

$$p_*^{\,SIS} = \frac{(\rho c)_L p_R + (\rho c)_R p_L + (\rho c)_L (\rho c)_R (u_L - u_R)}{(\rho c)_L + (\rho c)_R}$$

$$u_*^{\,SIS} = \frac{(\rho c)_L u_L + (\rho c)_R u_R + (p_L - p_R)}{(\rho c)_L + (\rho c)_R}$$

Slopes and Limiters

- Minmod

$$\min\mathrm{mod}(s_L,s_R)=\frac{1}{2}[\mathrm{sgn}(s_L)+\mathrm{sgn}(s_R)]\min(|s_L|,|s_R|)$$

$$\equiv median(s_L,s_R,0)$$

- Extended Minmod(*)

$$xm(s_L,s_R)=median(s_L,s_R,-(s_L+s_R))$$

$$=-(s_L+s_R)+\min\mathrm{mod}(2s_L+s_R,2s_R+s_L)$$

- Van Leer

$$vanLeer(s_L,s_R)=\min\mathrm{mod}(\frac{s_L+s_R}{2},\ 2s_L,\ 2s_R,\ 0)$$

* Huynh, H.T., SIAM J. Num.Anal. v32 5(1995)1565

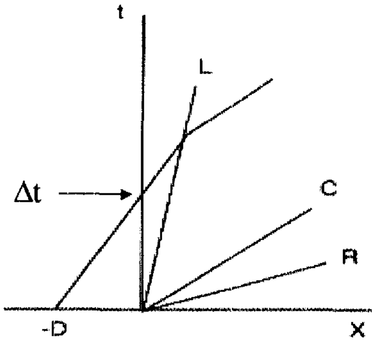
Each face, between two cells, constucts a linearly interpolated average from the low and high values passed to it from the cells, using the distances to the high and low cells' centers-of-mass. This face value is then passed back to the cells. (Multiple face values may get area-weighted if at a T-cell (jump in zonesize.)

The cells now calculate a slope between cell_lo and cell-center, $s_{L,}$ and cell-center and cell-hi, $s_{R.}$

The "iterated" minmod method goes through the above logic twice; on the second pass, the slope from the first (minmod) pass is used to give the face a modified hi or low cell-value.

All methods except for minmod tend to give comparable and sharper contacts in shock-tube problems. They also tend to give too much vorticity in 2-d shock-tube problems (by changing the form of the leading order truncation error from simple diffusion to something else), creating too "fractal" a pattern in the roll-ups of mushrooms. (See figures later on.)

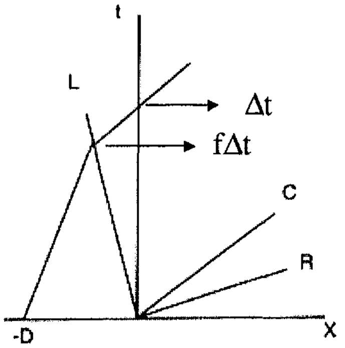
Advection -- two simple cases



Supersonic flow: dependence on donor cell only.

$$-D+\ (-Du_x+u_L^{1/2})\Delta t=0$$

$$\left\langle\frac{D}{\Delta t}\right\rangle=\frac{u_L^{1/2}}{1+u_x\Delta t}$$



Subsonic flow: dependence on both donor cell and contact.

$$-D+\ (-Du_x+u_L^{1/2})f\Delta t=(u_L^{1/2}-c_L^{1/2)})f\Delta t$$

$$(u_L^{1/2}-c_L^{1/2)})f\Delta t+(u_C-Du_x)(1-f)\Delta t=0$$

$$\Rightarrow f^2(u^*-u_L)u_x\Delta t$$

$$+f\left[u^*-u_L+s_L+(s_L-u^*)u_x\Delta t\right]-u^*=0$$

There are two other cases that mirror these two. In all cases, one can see that only one of the two zones is a "donor", and only one is a "donee".

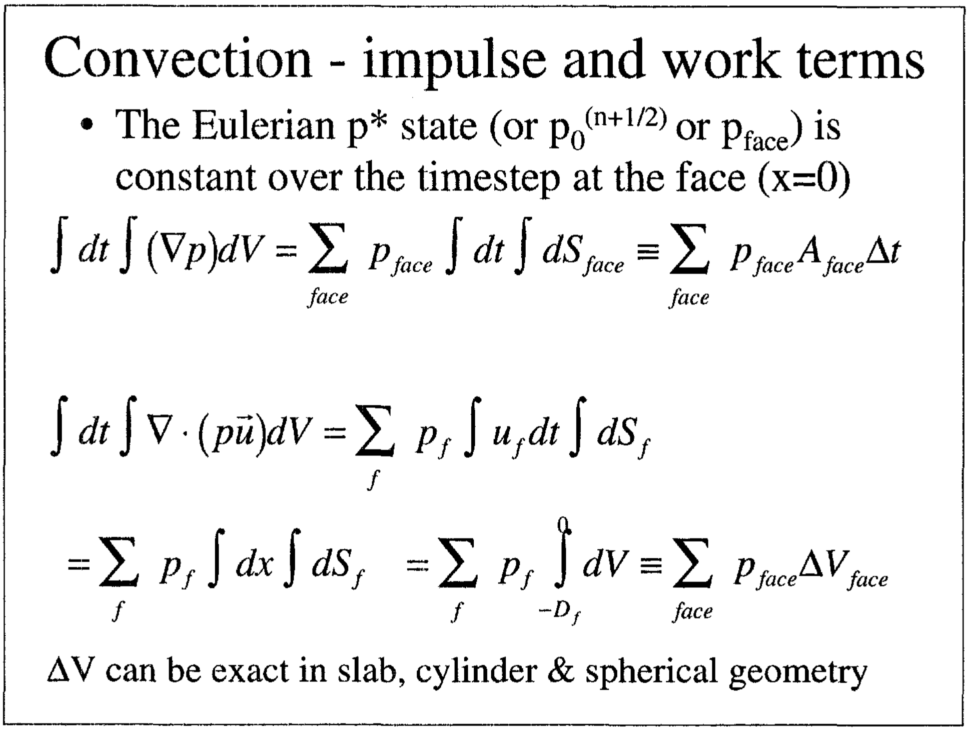
Rather than calculate

$$\Phi_R^{n+1}=\Phi_R^n+\Delta t\int d\vec{S}\cdot(\vec{u}\phi)^*$$

which doesn't offer a clear path forward for advecting Erad, e.g., we use

$$\Phi_R^{n+1}=\Phi_R^n+\int_{-D}^{0}\phi_L^n dV$$

$$where\quad \Phi=\int_{Vcell}\phi dV$$



Convection - impulse and work terms

- The Eulerian p* state (or $p_0^{(n+1/2)}$ or p_{face}) is constant over the timestep at the face (x=0)

$$\int dt \int (\nabla p) dV = \sum_{face} p_{face} \int dt \int dS_{face} \equiv \sum_{face} p_{face} A_{face} \Delta t$$

$$\int dt \int \nabla \cdot (p\vec{u}) dV = \sum_{f} p_f \int u_f dt \int dS_f$$

$$= \sum_{f} p_f \int dx \int dS_f \quad = \sum_{f} p_f \int_{-D_f}^{0} dV \equiv \sum_{face} p_{face} \Delta V_{face}$$

ΔV can be exact in slab, cylinder & spherical geometry

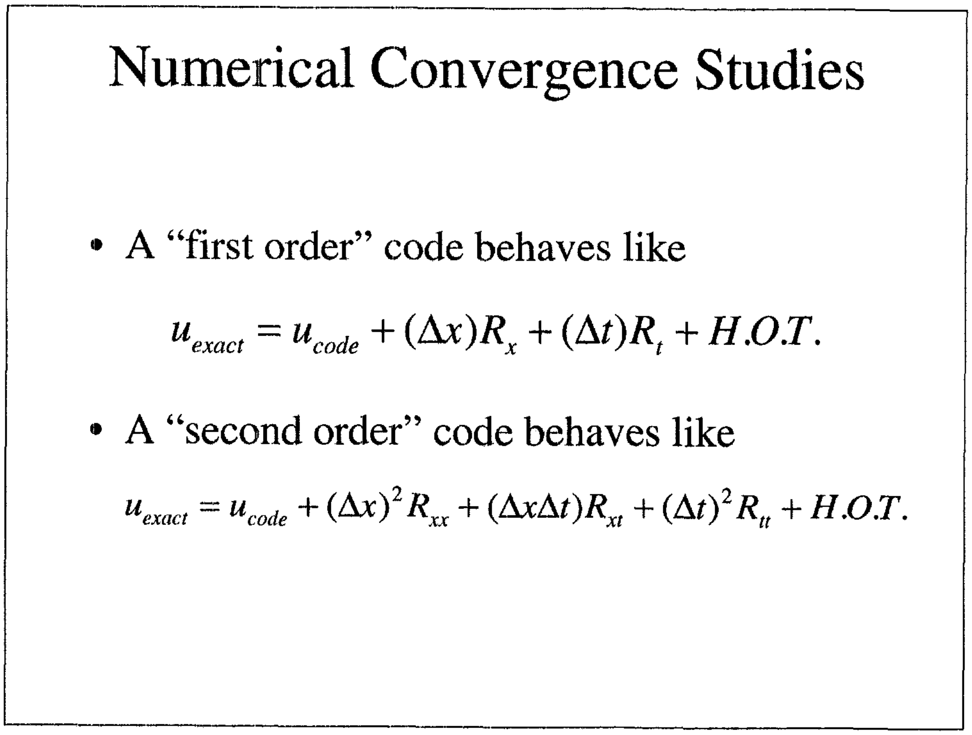
In our Lagrangian-flavored method, the pressure experienced at the lagrangian mass point is transferred to the Eulerian face, on the assumption that the pressure is constant between that point and the face at any instant of time.

Thus, what is P_{face} in the expression above is the time-average value of the pressure at the Lagrangian point:

$$\bar{p} = f \quad p_{fluid} + (1-f) \quad p_{Riemann}$$

$$= p_{Riemann}^{Lagrangian} - f\left(p_{Riemann}^{Lagrangian} - p_{fluid}\right)$$

In the limiting case of no velocity gradient, one can show that this Lagrangian average is equivalent to the expression that would result from calculating the true Eulerian, face- and time-centered Riemann state with the single intermediate state approximate Riemann solver.



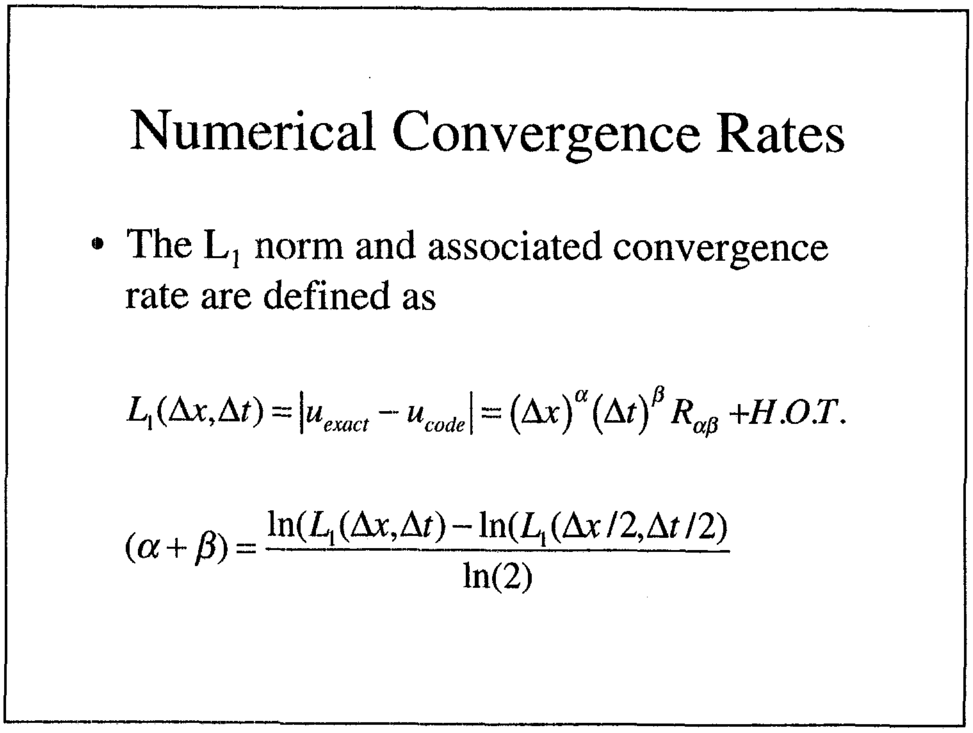
Numerical Convergence Studies

- A "first order" code behaves like

$$u_{exact} = u_{code} + (\Delta x)R_x + (\Delta t)R_t + H.O.T.$$

- A "second order" code behaves like

$$u_{exact} = u_{code} + (\Delta x)^2 R_{xx} + (\Delta x \Delta t)R_{xt} + (\Delta t)^2 R_{tt} + H.O.T.$$



Numerical Convergence Rates

- The L_1 norm and associated convergence rate are defined as

$$L_1(\Delta x, \Delta t) = \left| u_{exact} - u_{code} \right| = (\Delta x)^{\alpha} (\Delta t)^{\beta} R_{\alpha\beta} + H.O.T.$$

$$(\alpha + \beta) = \frac{\ln(L_1(\Delta x, \Delta t) - \ln(L_1(\Delta x/2, \Delta t/2)}{\ln(2)}$$

Convergence Examples

- Advection of a parabolic profile:
  - 2nd order in density, exact in velocity
- Adiabatic compression of a uniform sphere:
  - 2nd order in velocity, exact in density
- A linear wave in a bucket:
  - 2nd order in density and velocity
- A non-linear wave (Burgers' equation):
  - ~ 2nd order in density and velocity
- Shock Tubes
  - Sod prime (RCS): ~1st order
  - Kamm's Riemann tests (SCS): ~ 1st order
  - Kamm's Riemann tests (RCVCR): ~1.5- 2nd order if $t_0 > 0$
- 2-d Wave in a bucket: 2nd order
- 2-d Rider #0
- 2-d Benjamin shock tube -- effects of limiters on rollups !

Parabolic Advection - setup

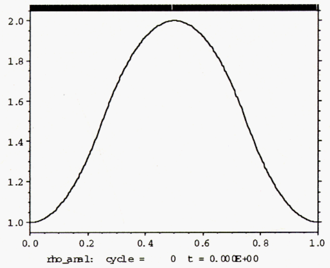
$$\rho(x,0) = 1 + 8(x - \frac{1}{4})^2 \qquad 0 < x < \frac{1}{2}$$

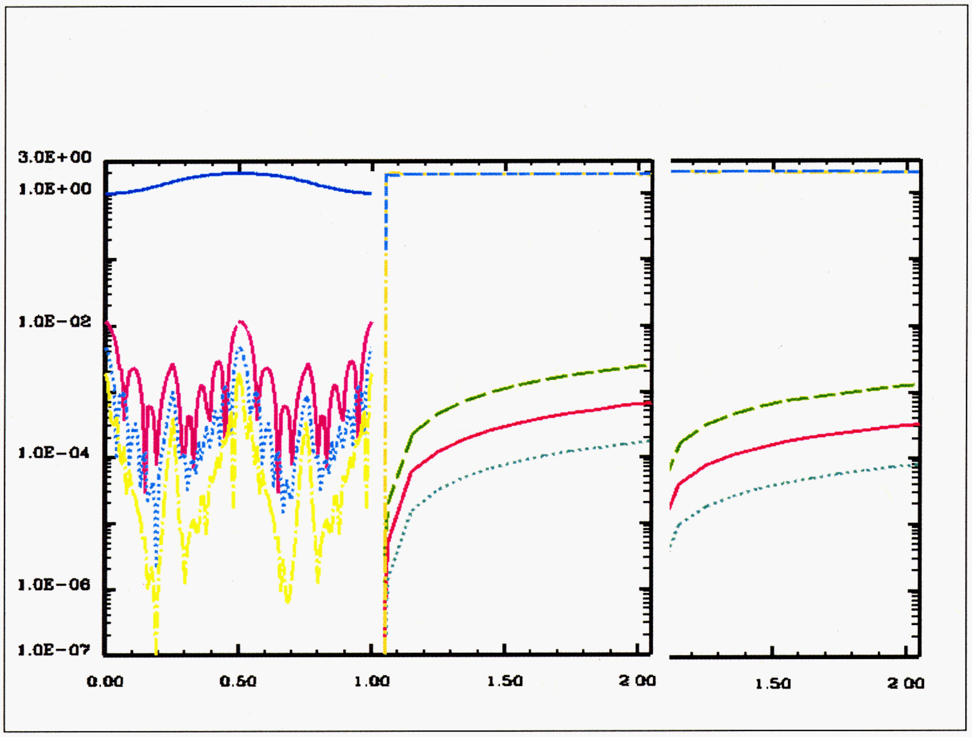
$$= 2 - 8(x - \frac{3}{4})^2 \qquad \frac{1}{2} < x < 1$$

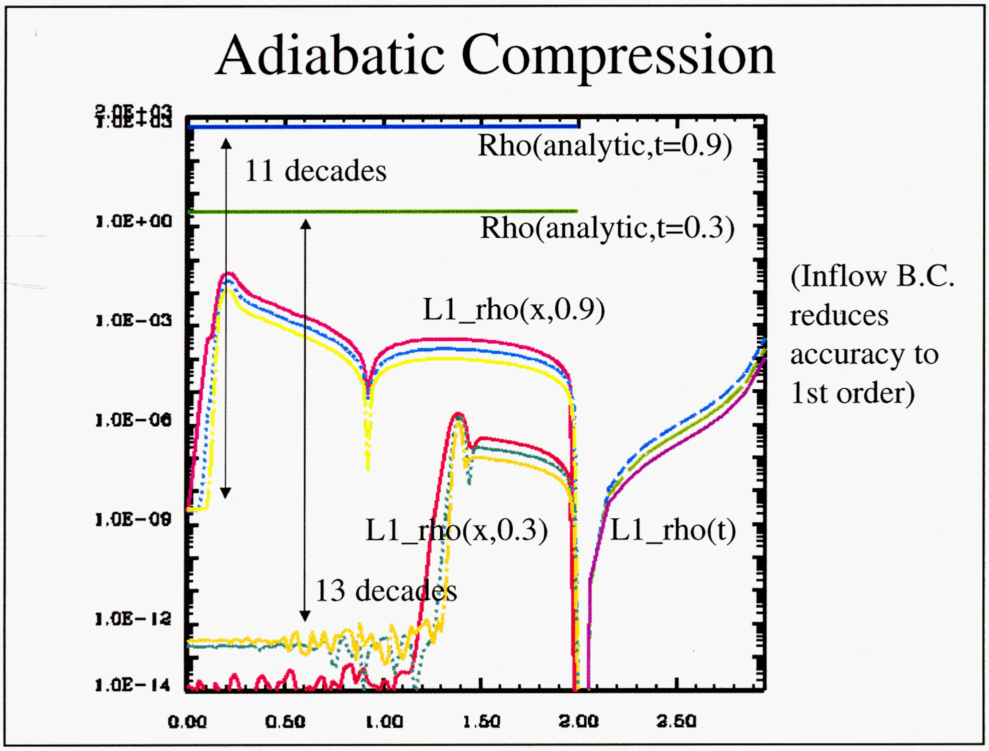
$$u(x,0) = 1$$

$$p(x,0) = (\gamma - 1)\rho e = 1$$

$$\gamma = \frac{7}{5} \qquad ;C_V = 1$$







Adiabatic Compression

(Inflow B.C. reduces accuracy to 1st order)

Adiabatic Compression (spherical setup)

$$\rho_0 = 1 \ ; \ e_0 = 0.01 \ ; \ \vec{u}_0 = -\vec{r} \ ; \ \gamma = 3/2$$

$$\rho(t) = \frac{\rho_0}{(1-t)^3} \quad ; \quad e(t) = \frac{e_0}{(1-t)^{3(\gamma-1)}}$$

$$\vec{u}(\vec{r},t) = \frac{-\vec{r}}{(1-t)}$$

Linear Wave in a Bucket - setup

$$\rho_0 = 1 \ ; \ u_0 = 0 \ ; \ e_0 = 0.9 \ ; \ \gamma = 5/3$$

$$(\ p_0 = 0.6 \ ; \ c_s = 1 \ )$$

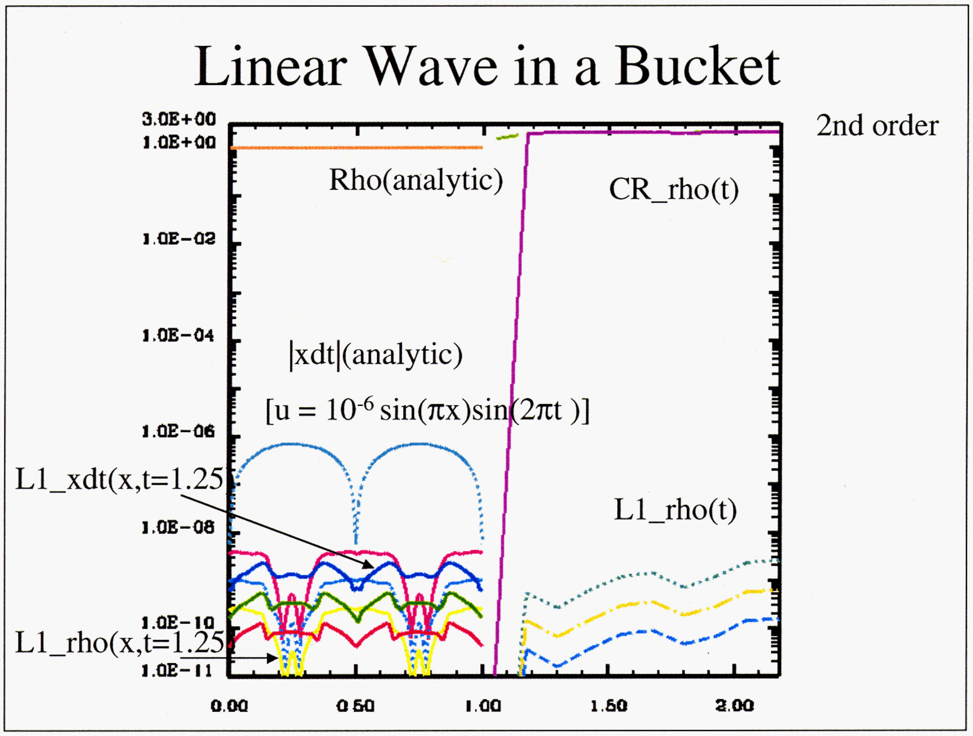
$$\rho_1 = 10^{-6}\cos(2\pi x)$$

$$u_1 = 0$$

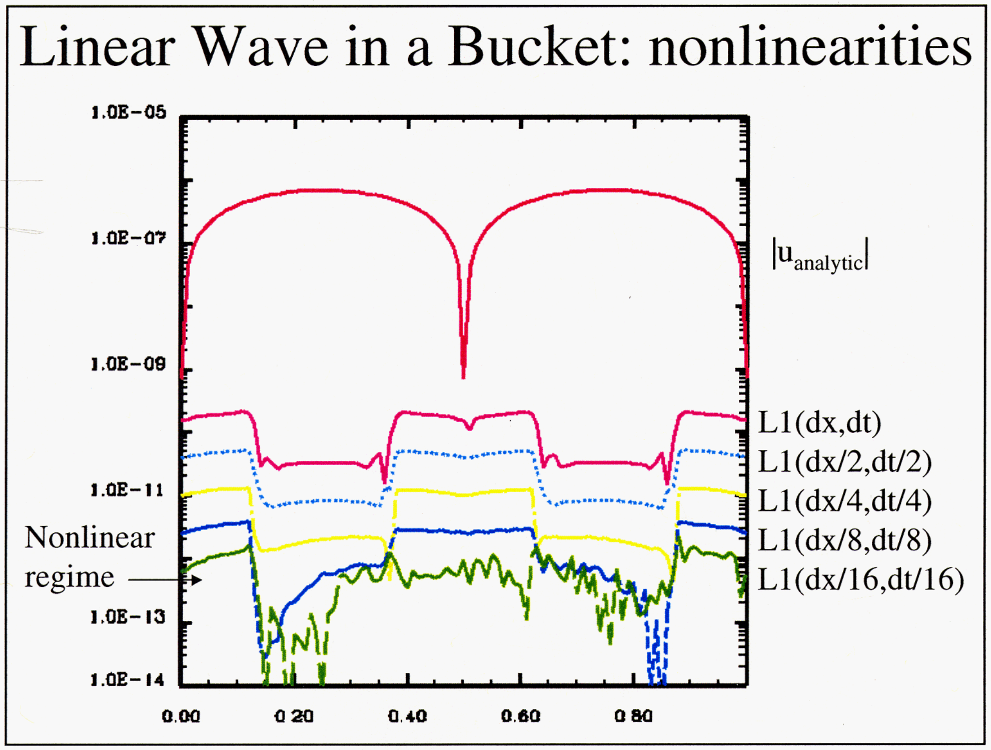
$$e_1 = \frac{p_0}{\rho_0}\left(\frac{10^{-6} + \rho_1}{\rho_0 + \rho_1}\right)$$

$$\frac{\partial^2 \phi_i}{\partial t^2} + \frac{\partial^2 \phi_i}{\partial x^2} = 0 \qquad ; \ \phi_i = (\rho_1, u_1, e_1)$$

Linear Wave in a Bucket

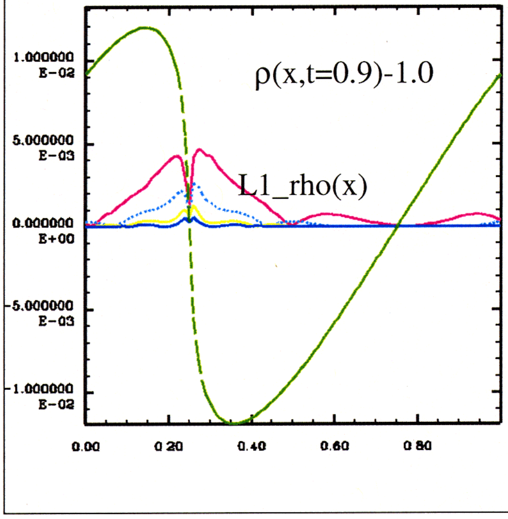


Linear Wave in a Bucket: nonlinearities

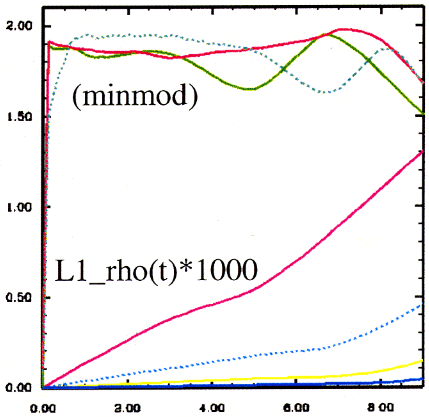


Nonlinear Wave (Burgers')

With appropriate sinusoidal initial conditions: $\frac{d\hat{u}}{dt}+\frac{d\hat{u}^2}{dx}=0$



Convergence rate = 1.9->1.5



Nonlinear Wave - Burgers' Eqn setup

$$\frac{u_0}{u_\infty}=M_\infty+\frac{\cos(2\pi x)}{(\gamma+1)\pi \mathrm{T}_{shock}} \quad ; \quad \frac{a_0}{a_\infty}=1+\frac{\gamma-1}{\gamma+1}\frac{\cos(2\pi x)}{2\pi \mathrm{T}_{shock}}$$

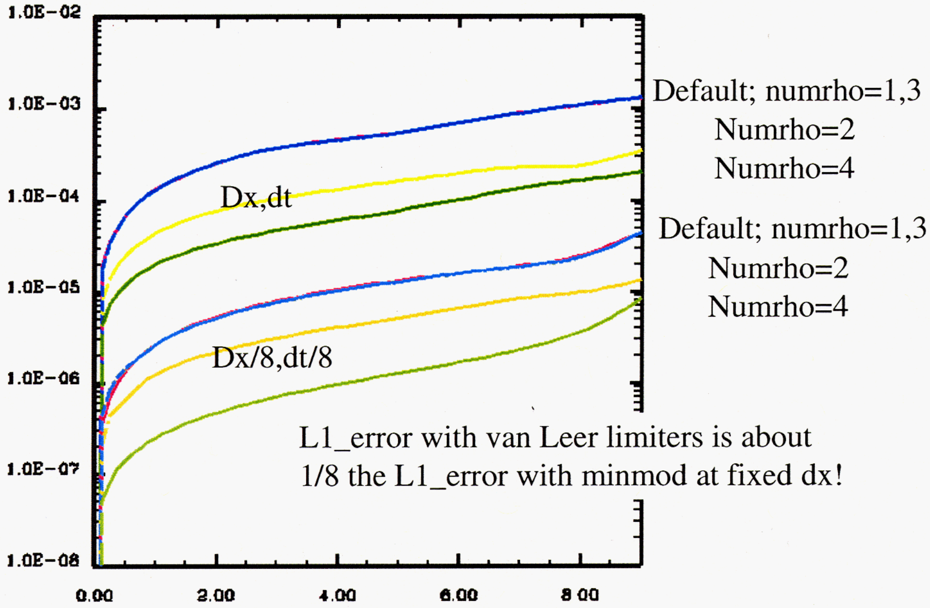
$$T_{shock}=10 \;;\; \frac{p_0}{p_\infty}=\left(\frac{a_0}{a_\infty}\right)^{\frac{2\gamma}{\gamma-1}} \;;\; \frac{\rho_0}{\rho_\infty}=\left(\frac{a_0}{a_\infty}\right)^{\frac{2}{\gamma-1}} \;;\; \gamma=5/3 \;;\; M_\infty=0$$

With the change of variable: $\tilde{u}=\frac{\gamma+1}{2}u(x)+a_\infty-\frac{\gamma-1}{2}u_\infty$

all three Euler equations reduce to $\frac{\partial\tilde{u}}{\partial t}+\frac{\partial}{\partial x}\left(\frac{\tilde{u}^2}{2}\right)=0$

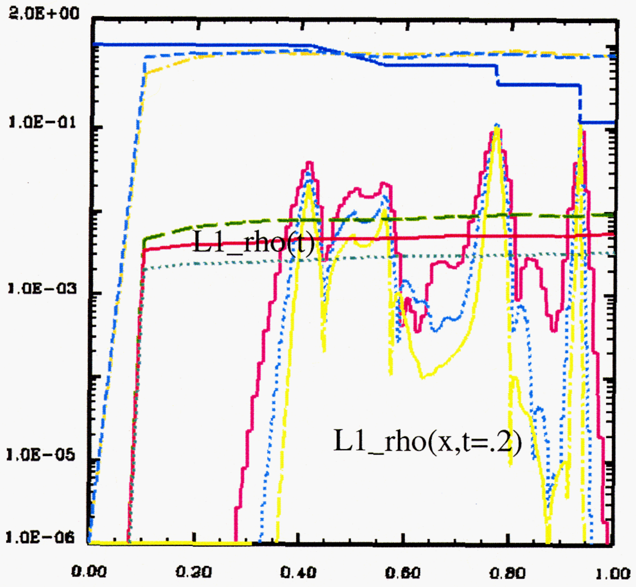
with solution $\tilde{u}(x,t)=\tilde{u}^0(x-\tilde{u}(x,t)\cdot t)$

Nonlinear Wave (Burgers): L1_xdt(t)



L1_error with van Leer limiters is about
1/8 the L1_error with minmod at fixed dx!

Shock tubes - Sod prime L1(rho)



CR=0.75

(Numrho=1)

Shock Tubes - Sod prime

$x_{face} = 1/2 \quad ; \quad \gamma = 7/5$

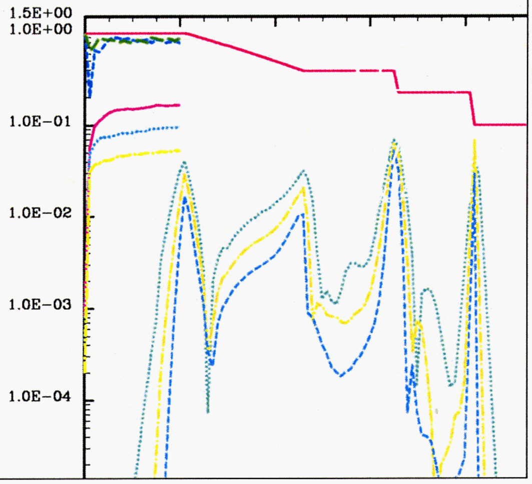
$(\rho, u, p)_L = (1, \ 3/4, \ 1)$

$(\rho, u, p)_R = (1/8, \ 0 \ 1/10)$

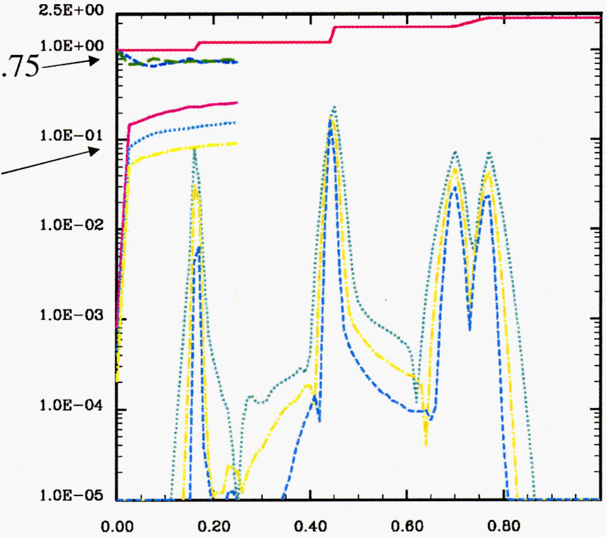
$\Rightarrow u_L - c_L = -0.43 \ ; \ u_{L*} - c_{L*} = +0.29$

$[original \ \ Sod : (\rho, u, p)_L = (1, \ 0, \ 1)$

$\Rightarrow u_L - c_L = -1.18 \ ; \ u_{L*} - c_{L*} = -0.07]$



Shock Tubes - Kamm's SCR

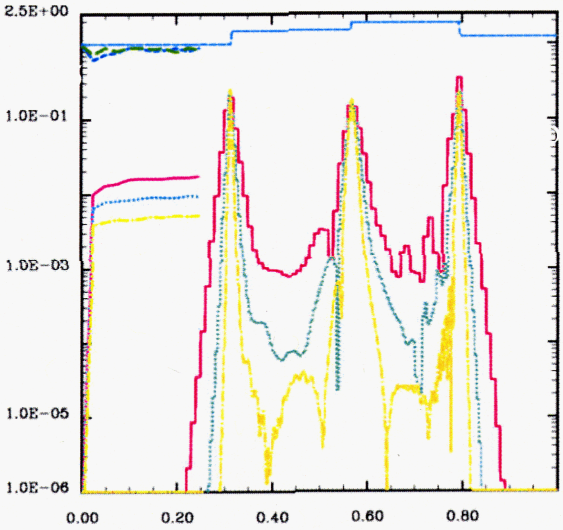


Rho(x,t=.25)

CR_rho(t) ~0.75

L1_rho(t)*20

L1_rho(x):
Dx, dt
Dx/2,dt/2
Dx/4,dt/4

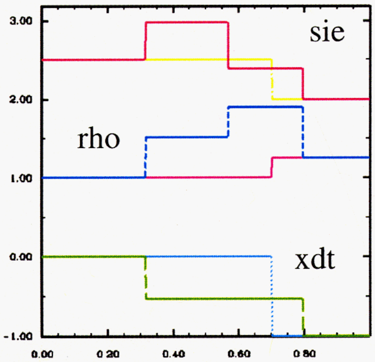


Shock Tubes - Kamm’s SCS

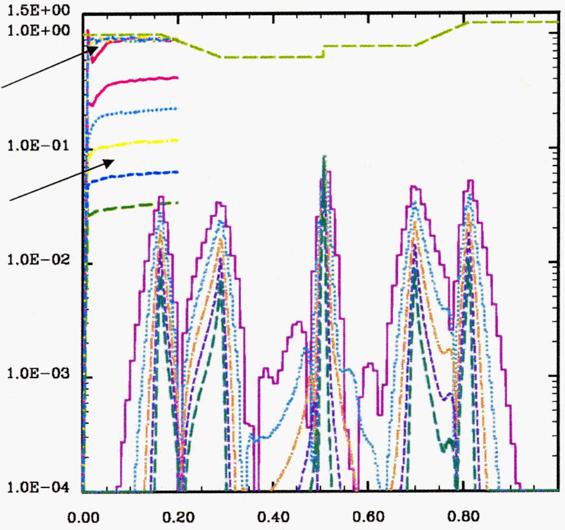
$x_{face} = 0.7$; $\gamma = 7/5$

$(\rho, u, p)_L = (1,\ \ 0\ \ \ 1)$

$(\rho, u, p)_R = (5/4,\ \ -1\ \ \ 1)$



Shock Tubes - Kamm's RCR

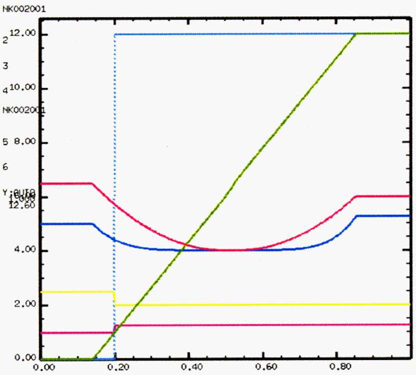


CR_rho(t) ~0.9

L1_rho(t)*40

Rho(x,t=.2)

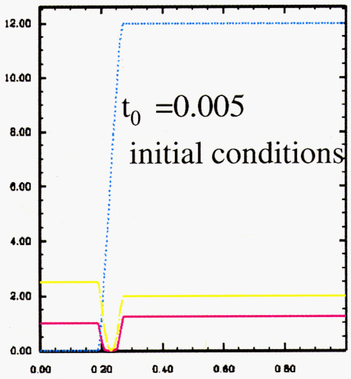
L1_rho(x,.2)
Dx, dt
Dx/2,dt/2
…
Dx/16,dt/16



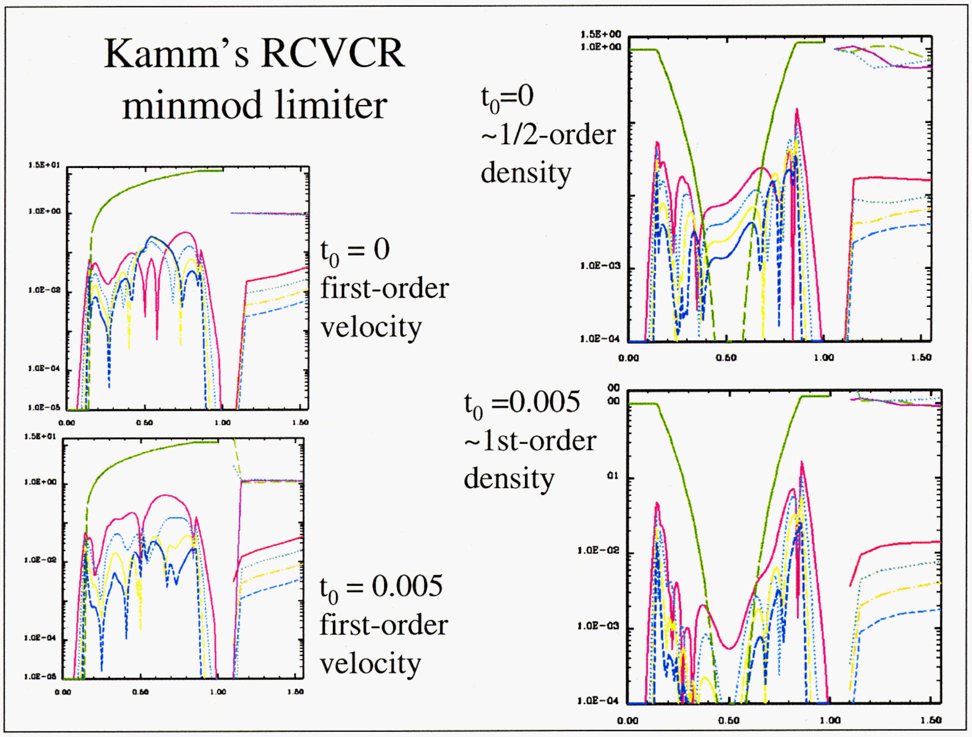
Xdt(0,0.05)

Sie(0.05)
Rho(0.05)
(Offset by 4.)

Sie(0)
Rho(0)



t_0 =0.005
initial conditions



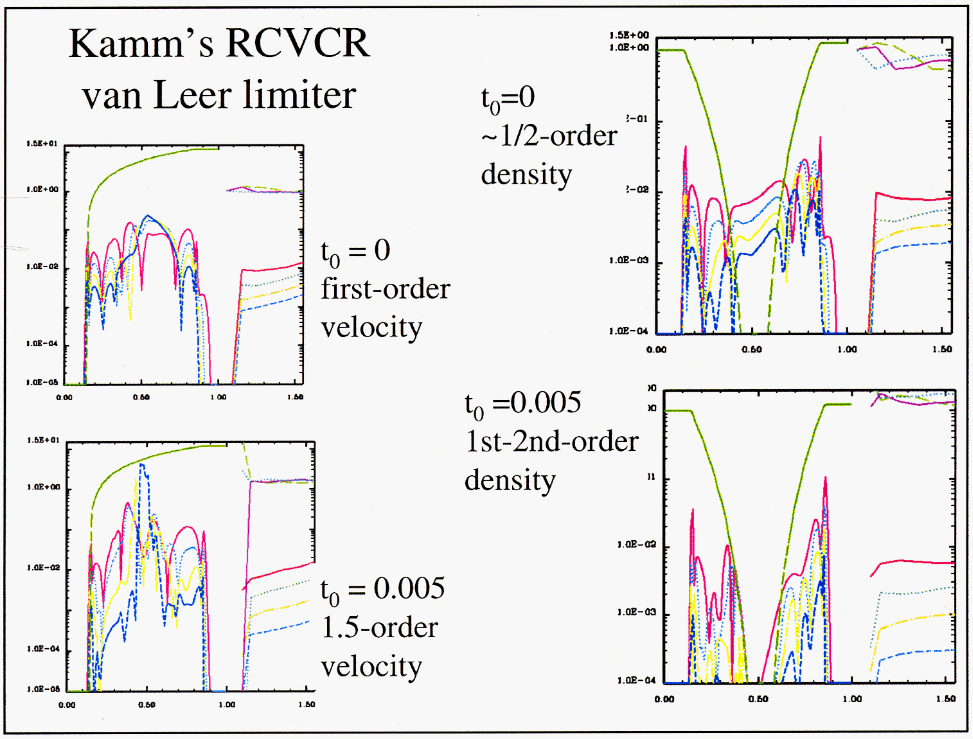
Kamm’s RCVCR minmod limiter

$t_0 = 0$ first-order velocity

$t_0 = 0.005$ first-order velocity

t_0=0 ~1/2-order density

t_0 =0.005 ~1st-order density



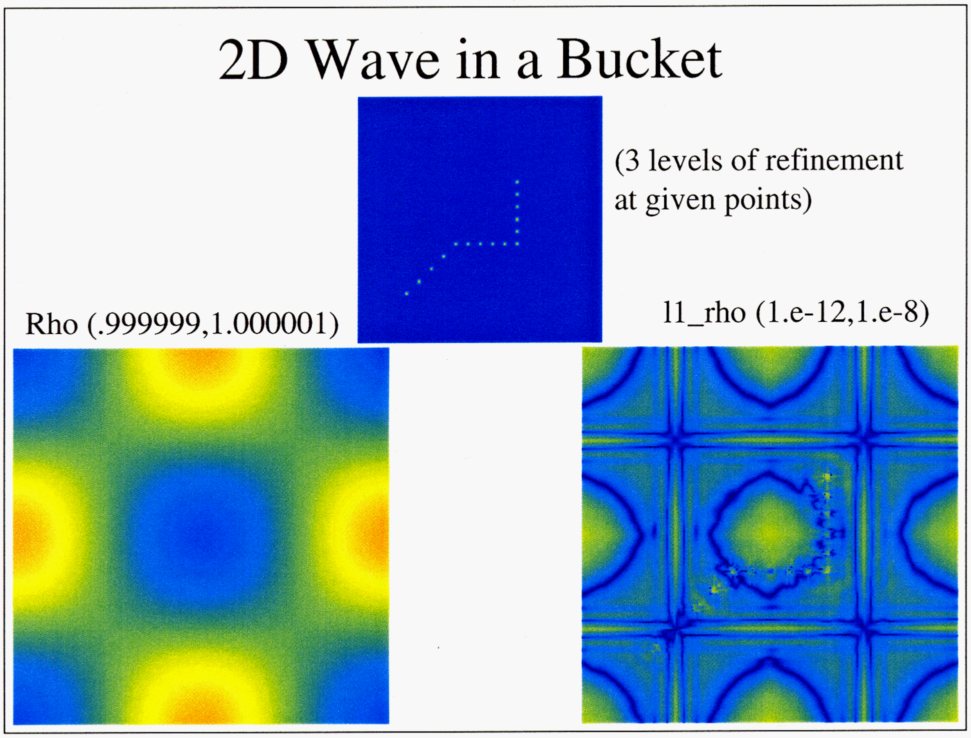
Kamm’s RCVCR van Leer limiter

$t_0 = 0$ first-order velocity

$t_0 = 0.005$ 1.5-order velocity

t_0=0 ~1/2-order density

t_0 =0.005 1st-2nd-order density



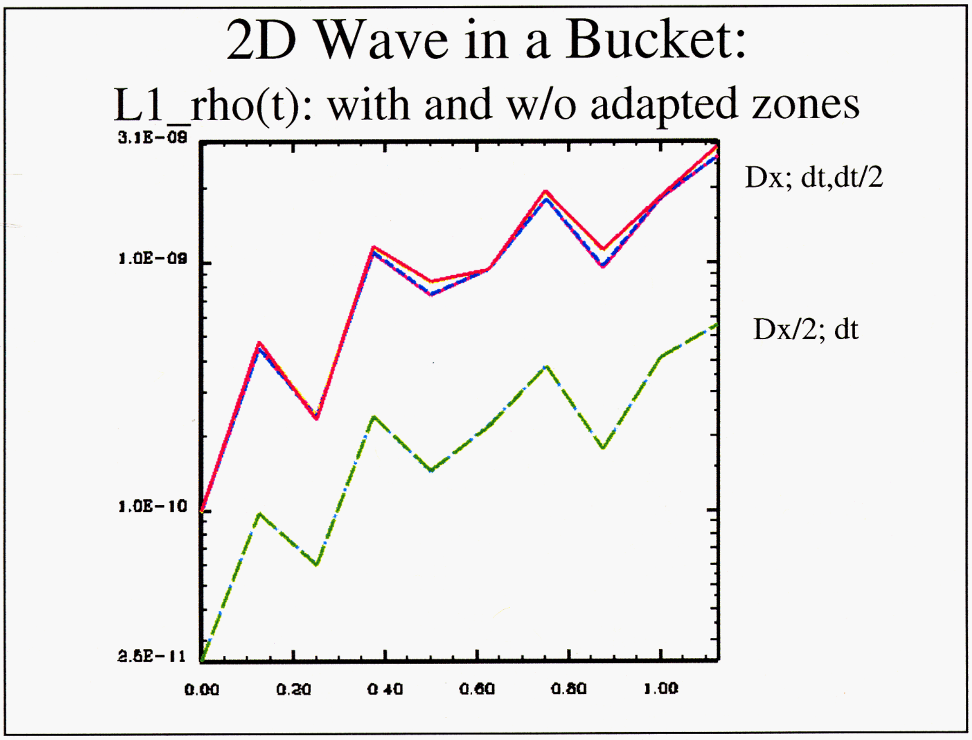
2D Wave in a Bucket

(3 levels of refinement at given points)

Rho (.999999,1.000001)

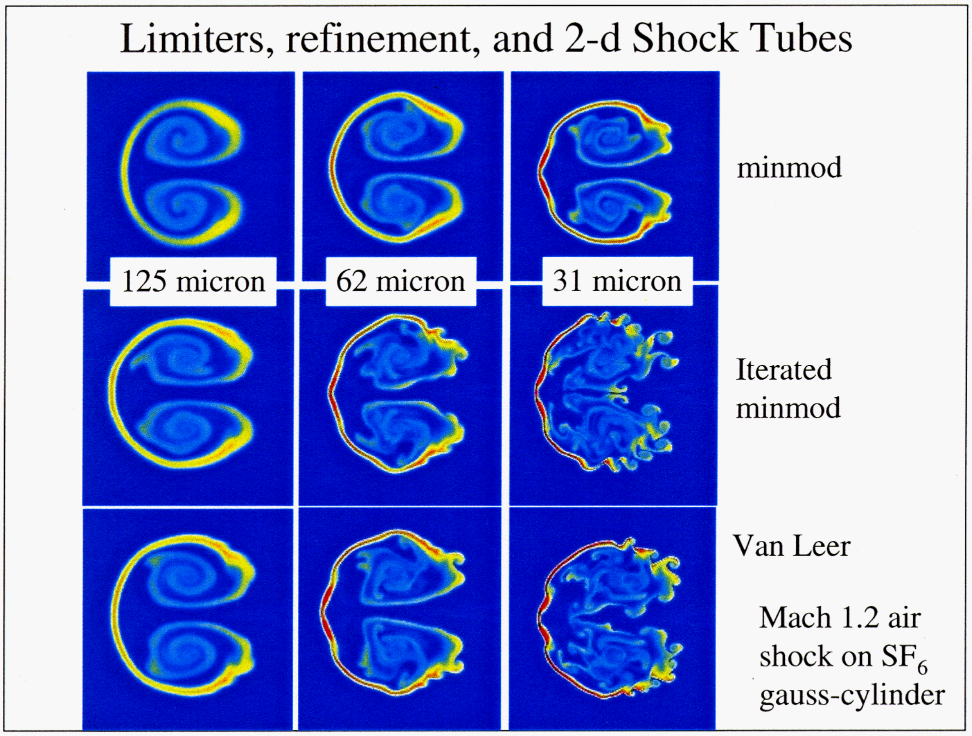
l1_rho (1.e-12,1.e-8)

On other cycles, the imprint in the L1-norm is even less obvious.



2D Wave in a Bucket:

L1_rho(t): with and w/o adapted zones



Limiters, refinement, and 2-d Shock Tubes

125 micron

62 micron

31 micron

minmod

Iterated minmod

Van Leer

Mach 1.2 air
shock on SF_6
gauss-cylinder